	<p style="text-align: center;">Metody programowania 2023/2024 Pakowanie plecaka</p>	<p style="text-align: center;">P_06</p>
---	---	---

## Opis

Napisz aplikację w Javie zawierającą dwie funkcje: rekurencyjną *recPackage (...)* oraz jej iteracyjną wersję *StackSimPackage (...)*, wykorzystującą stos i będącą symulacją funkcji rekurencyjnej, rozwiązujące problem pakowania plecaka opisany na wykładzie, to znaczy znajdujący sekwencję elementów, których sumaryczna waga jest dokładnie równa pojemności plecaka (wybrane elementy całkowicie wypełniają plecak).

Aplikacja powinna umożliwiać operowanie na plecaku o zadanej pojemności oraz zadanej liczbie elementów o różnych wagach umieszczanych w plecaku.

## Wejście

Dane do programu wczytywane są ze standardowego wejścia (klawiatury) zgodnie z poniższą specyfikacją:

1. Pierwszą podawaną wartością będzie dodatnia liczba całkowita  $z$  ( $1 \leq z \leq 10^6$ ), oznaczająca ilość zestawów danych.
2. Każdy zestaw danych zawiera w kolejnych wierszach:
  - liczbę całkowitą  $n$  ( $1 \leq n \leq 10^6$ ) oznaczająca pojemność plecaka.
  - liczbę całkowitą  $k$  ( $1 \leq k \leq 10^6$ ) oznaczająca liczbę elementów mogących wypełnić plecak.
  - niepowtarzające się liczby całkowite  $a_1, \dots, a_k$  ( $1 \leq a_i \leq 10^6$  dla  $i$  od 1 do  $k$ ), będące wagami kolejnych elementów.


## Wyjście

W przypadku braku poszukiwanego rozwiązania program wypisuje słowo **BRAK**, natomiast dla każdego zestawu danych z istniejącym rozwiązaniem program wypisuje w dwóch liniach kolejno:

```
REC:  pojemność plecaka =
ITER: pojemność plecaka =
```

następnie w obu liniach wypisuje ciąg wag wybranych elementów oddzielonych pojedynczą spacją, z zachowaniem wejściowej kolejności i dający w sumie zadaną pojemność plecaka.

Przy czym spośród wszystkich ciągów, wypisywany ciąg wag musi być pierwszym względem porządku leksykograficznego numerów swoich elementów.

	<p style="text-align: center;"><b>Metody programowania 2023/2024</b> <b>Pakowanie plecaka</b></p>	<p style="text-align: center;">P_06</p>
---	---	---

## Wymagania implementacyjne

1. // Imie Nazwisko – nr grupy
2. Jedynym dozwolonym importem jest obsługa wczytywania z klawiatury, to jest:  

```
import java.util.Scanner;
```
3. Główna klasa musi nazywać się *Source*, co oznacza ogólne ramy kodu postaci:  

```
class Source {
    public static void main( String [] args ) {
        ...
    }
}
```
4. Wczytywanie musi się odbywać przez pojedynczą zmienną klasy *Scanner*, zadeklarowaną zewnętrznie w stosunku do wszystkich metod głównej klasy.  
W praktyce oznacza to tylko jedną deklarację w przykładowej postaci, np.:  

```
public static Scanner sc = new Scanner(System.in);
```

w pierwszej linii ciała głównej klasy.
5. Kod programu powinien zawierać komentarze i testy zgodnie z opisem podanym w Regulaminie zaliczania programów na BaCy.
6. Funkcja rekurencyjna nie może zawierać pętli.

## Przykład danych

Wejściowe:	Wyjściowe:
2 20 5 11 8 7 6 5 21 3 5 6 7	REC: 20 = 8 7 5 ITER: 20 = 8 7 5 BRAK